I hereby certify that this is being deposited with the U.S. Postal Service "Express Mail Post Office to Addressee" service under 37 CFR § 1.10 on the date indicated below and is addressed to:

Assistant Commissioner for Patents
Box Patent Application
Washington, D.C. 20231

By: Teresa A. Fleming

Typed Name: Teresa A. Fleming

Express Mail Label No.: EL 828142456 US

Date of Deposit: 7/13/01

Attorney Docket No.: SUN-P5494-MDF

METHOD FOR GENERATING CONFIGURATION TABLES AND FOR FORWARDING PACKETS THROUGH A NETWORK

Inventor: James M. Avery

1. FIELD OF THE INVENTION

The present invention generally relates to hierarchical networks. More specifically, the present invention relates methods of generating configuration tables and forwarding packets through a hierarchical network.

2. BACKGROUND

In an effort to increase I/O bandwidth in high performance processor based systems, a number of companies have developed the HyperTransport ("HT") I/O interconnect structure. Briefly, the HT I/O interconnect structure is a scalable device level architecture that provides a significant increase in transaction throughput over existing I/O bus architectures such as Peripheral Component interconnect ("PCI") and Advanced Graphics Port ("AGP").

The foundation of the HT I/O interconnect is dual point-to-point unidirectional links consisting of a data path, control signals, and clock signals. The HT I/O interconnect can provide both point-to-point links and a scalable network topology using HT I/O switching fabrics. Thus, an HT based system can be expanded using HT switches to support multilevel, highly complex systems.

Communications between multiple HT I/O devices are known as data streams. Each data stream contains one or more packets of information. Each packet of information contains a packet ID and a data payload. The packet ID is also commonly referred to as a unit ID. Because all packets are transferred to or from a host bridge, the packet ID provides information that can be utilized to determine the source or destination of the packet. A more detailed description of the HT I/O interconnect structure is presented in Appendix A.

While HT switches that handle multiple HT I/O data streams and manage the interconnection between attached HT I/O devices have been proposed in concept, no methods for managing configuration of such switches are known. Similarly, no methods for efficiently forwarding packets through such switches are known. Thus, a need exists for efficiently managing configuration of and packet forwarding through HT switches.

3. SUMMARY OF INVENTION

One embodiment of the invention is a method, performed by a computer system that includes a host processor coupled to a first bus, a first switch coupled to the first bus and a second bus, a second switch coupled to the second bus and a third bus, and a device coupled to the third bus. The method of storing information in a configuration register

includes: issuing a first configuration transaction onto the first bus; forwarding the first configuration transaction to the second bus; translating the first configuration transaction into a second configuration transaction; forwarding the second configuration to the third bus; and storing information in the configuration register.

Another embodiment of the invention is a method, performed by a computer system that includes a host processor coupled to a first bus, a first switch coupled to the first bus and a second bus, a second switch coupled to the second bus and a third bus, and a device coupled to the third bus. The method of retrieving information from a configuration register includes: issuing a first configuration transaction onto the first bus; forwarding the first configuration transaction to the second bus; translating the first configuration transaction into a second configuration transaction; forwarding the second configuration transaction to the third bus; and retrieving information from the configuration register.

Another embodiment of the invention is a method, performed by a computer system that includes a host processor coupled to a bus and a switch coupled to the bus. The method includes: issuing a configuration transaction that includes a primary-segment field and includes a secondary-segment field onto the bus; and receiving the configuration transaction in the switch.

Another embodiment of the invention is a method, performed by a computer system that includes a host processor coupled to a bus, and a switch coupled to the bus. The method generates a configuration-forwarding table by: detecting the presence of the switch; determining the number of primary ports present in the switch; for each primary port present in the switch, determining if the primary port is enabled or disabled; and for

each enabled primary port in the switch, storing a value in the configuration-forwarding table that identifies the primary segment number of the bus that is coupled to the port.

Another embodiment of the invention is a method, performed by a computer system that includes a host processor coupled to a first bus, and a switch coupled to the first bus, of generating a configuration-forwarding table. The method includes: detecting the presence of the switch; determining the number of secondary ports present in the switch; for each secondary port present in the switch, determining if the secondary port is enabled or disabled; and for each enabled secondary port in the switch, storing a value in the configuration-forwarding table that identifies the secondary segment number of the bus that is coupled to the port.

Another embodiment of the invention is a method, performed by a computer system that includes a host processor coupled to a first bus, a first switch coupled to the first bus and a second bus, and a second switch coupled to the second bus. The method of forwarding a configuration transaction includes: issuing a type1 configuration transaction on the first bus; receiving the type1 configuration transaction in the first switch; evaluating a logical equation; if the result of the evaluation of the logical equation is a first value, then forwarding the type1 configuration transaction to the second switch.

Still another embodiment of the invention is a method, performed by a computer system that includes a host processor that is coupled to a first bus, and a switch that is coupled to the first bus. The method of forwarding a packet includes: receiving a packet, determining the Unit ID of the packet; retrieving a primary-segment value from a first storage location within the switch; retrieving a secondary-segment value from a second

storage location within the switch; forwarding the packet through a port that is coupled to a bus that is identified by the primary-segment value and the secondary-segment value.

4. BRIEF DESCRIPTION OF THE FIGURES

Figure 1 presents a computer system.

Figure 2 presents one embodiment of a configuration-forwarding table.

Figure 3 presents another computer system.

Figure 4 presents another embodiment of a configuration-forwarding table.

Figure 5 presents another computer system.

Figure 6 presents one embodiment of a packet-forwarding table.

Figure 7 presents a portion of a computer system.

Figure 8 presents another embodiment of a packet-forwarding table.

Figure 9 presents one method of storing information in a configuration register in a device.

Figure 10 presents one method of retrieving information from a configuration register in a device.

Figure 11 presents one method of issuing a configuration transaction.

Figure 12 presents one method of generating a configuration-forwarding table.

Figure 13 presents another method of generating a configuration-forwarding table.

Figure 14 presents one method of forwarding a configuration transaction.

Figure 15 presents another method of forwarding a configuration transaction.

Figure 16 presents one method of forwarding a packet.

5. DESCRIPTION OF THE PREFERRED EMBODIMENTS

The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed embodiments may be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features disclosed herein.

5.1 Computer System with a Plurality of HT I/O Devices and an HT I/O Switch

One embodiment of the invention is shown in Figure 1, which presents a computer system 100. The computer system 100 contains a host processor 110 that includes an HT I/O port 102 and a memory port 105. The HT I/O port 102 is coupled to a first HT I/O interconnect 120. The first HT I/O interconnect 120 is also coupled to a first HT I/O device 130, an HT switch 140, and a second HT I/O device 150. The HT switch 140 is also coupled to a third HT I/O device 160 and a fourth HT I/O device 170.

5.2 HT I/O Device

The first HT I/O device 130 contains a plurality of configuration registers. The plurality of configuration registers store information about the HT I/O device 130. A portion of these configuration registers may be similar to the configuration registers defined in the PCI Local Bus Specification, Revision 2.2. For example, the configuration

registers could store information such as the HT I/O device's vendor ID, device ID, device class, Base Address Registers ("BARs"), capabilities, and status. Some of the configuration registers in the first HT I/O device 130 may be read from and written to. However, other configuration registers may only be read from. For example, the configuration register that contains an HT I/O device's vendor ID is typically a read only register.

The second HT I/O device 150, the third HT I/O device 160, and the fourth HT I/O device 170 would also typically contain configuration registers such as those described above.

5.3 HT Switch

The HT switch 140 contains one primary port and two secondary ports (not shown). A primary port is a port that is closer to the host processor than other ports on the HT switch. A secondary port is a port that is further from the host processor than other ports on the HT switch. The HT switch's primary port is coupled to the first HT I/O interconnect 120. The HT switch's first secondary port is coupled, via segment 180, to the third HT I/O device 160. Similarly, the HT's second secondary port is coupled, via segment 190, to the fourth HT I/O device 170.

The HT switch 140 also contains a plurality of configuration registers. Some of the configuration registers are similar to the configuration registers described in the PCI-to-PCI Bridge Architecture Specification Revision 1.1. For example, the HT switch's configuration registers could contain the switch's vendor ID, device ID, device class, Base Address Registers (BARs), capabilities, and status. However, in addition, to the

above configuration registers, in some embodiments of the invention, an HT switch may also include a plurality of configuration-forwarding tables and a packet-forwarding table.

5.4 Primary Configuration-Forwarding Tables

A configuration-forwarding table may be associated with a specific HT switch port. In such embodiments, a configuration-forwarding table is used to forward configuration transactions from the port on an HT switch that has been associated with the configuration-forwarding table to another port on the HT switch. Such forwarding is discussed in more detail in Section 5.6.4 and Section 5.9.2.

Figure 2 presents one embodiment of a configuration-forwarding table 200. This configuration-forwarding table 200 is associated with the HT switch's primary port. Because the configuration-forwarding table 200 is associated with a primary port, the table may be referred to as a primary port configuration-forwarding table. Configuration-forwarding table 200 is used to forward configuration packets from the HT switch's primary port to one of the HT switch's secondary ports.

5.4.1 Capability ID

The first element in the configuration-forwarding table 200 is a capability ID 205. The capability ID 205 contains information that indicates that the switch is an HT switch.

5.4.2 Capabilities Pointer

The second element in the configuration-forwarding table 200 is a capabilities pointer 210. The capabilities pointer 210 may contain a pointer to a linked capabilities

list that contains HT specific capability registers. Alternatively, if the HT switch does not contain any additional capabilities, then the capabilities pointer 210 may be set to a value of 0. Because the capabilities of a device are device specific, the register that contains the capabilities pointer 210 is typically read only.

5

5.4.3 Command Bits

The third element in the configuration-forwarding table 200 contains command bits 215. A first group of the command bits may indicate the number of primary port(s) present in the HT switch. In such an embodiment, the first group of command bits would indicate that the HT switch 140 contains one primary port. A second group of command bits may indicate the number of secondary port(s) present in the HT switch. In such an embodiment, the second group of the command bits would indicate that the HT switch 140 contains two secondary ports.

The command bits 215 may also contain a bit that indicates if the configuration-forwarding table is associated with the HT switch's primary ports or is associated with the HT switch's secondary ports. As discussed in Section 5.4, the configuration-forwarding table 200 is associated with the HT switch's primary port. Thus, the above command bit would indicate that the configuration-forwarding table is a primary configuration-forwarding table.

The command bits 215 may also include a third group of bits that indicate the layout of the remainder of the configuration-forwarding table 200.

5.4.4 Configuration Transaction-Forwarding Data

The remainder of the configuration-forwarding table 200 contains configuration transaction-forwarding data. The configuration transaction-forwarding data is utilized by the computer system to forward configuration transactions, such as those discussed in Section 5.9.2, through an HT switch to devices coupled to the HT switch's secondary ports. These devices may be directly coupled to an HT switch. For example, as shown in Figure 3, the first HT switch 305 is directly coupled to the second HT switch 310 via segment 320. Similarly, segment 325 directly couples the second switch 310 to the first HT I/O device 315. Segments that directly couple a device to an HT switch may be referred to as primary segments with respect to that HT switch.

Devices may also be coupled to an HT switch indirectly. For example, the first HT switch 305 is indirectly coupled to the first HT I/O device 315 via segment 325. Segments that indirectly couple a device to an HT switch may be referred to as secondary segments with respect to that HT switch.

5.4.4.1 Number Range ("Match")

In one embodiment, the first configuration transaction-forwarding element, *P0 Match*, identifies the primary segment that is directly coupled to the first secondary port of an HT switch. In another embodiment, the first configuration transaction-forwarding element, *P0 Match*, indicates the number range of the highest numbered HT primary segment that is directly coupled to the first secondary port of an HT switch. As discussed in Section 5.8 and Section 5.9.2, the number range can be used to determine

when to respond to type1 configuration transactions on the primary port and forward them to secondary ports as type0 configuration transactions or as type1 configuration transactions.

5.4.4.2 Number Span ("Mask")

In one embodiment, the second configuration transaction-forwarding element, *P0 Mask*, indicates the span of the highest numbered HT primary segment that is coupled to primary port P0. As discussed in Section 5.9.2, the number span can be used to determine when to respond to type1 configuration transactions on the primary port and forward them to secondary ports as type 1 configuration transactions.

5.5 Secondary Configuration-Forwarding Tables

Figure 4 presents an embodiment of a configuration-forwarding table 400. Configuration-forwarding table 400 is associated with the HT switch's first secondary port. Because the configuration-forwarding table 400 is associated with a secondary port, the table may be referred to as a secondary port configuration-forwarding table. This table may be identical in structure to the primary configuration-forwarding table 200 discussed in Section 5.4. Second configuration-forwarding tables may be used to record the number range, *Sy Match*, and span, *Sy Mask*, of the highest numbered HT secondary segment that is connected to secondary port "y."

5.6 Configuration of the First HT Switch

In one embodiment of the invention, when the computer system 500, shown in Figure 5, is first powered on, software is executed by the host processor 505 that stores values to two temporary variables. In one embodiment, the value 0x1111 may be stored both in a first variable, *Temp-Primary-Segment* and a second variable, *Temp-Secondary-Segment*. Next, the software probes the first HT I/O interconnect 540 to determine if any HT I/O devices are coupled to the bus.

5.6.1 Type0 Configuration Transactions

One method of probing the HT I/O interconnect 540 is issuing a type0 configuration read transaction from the host processor 505 onto the HT I/O interconnect 540. Type0 configuration read transactions retrieve information from configuration registers that are located within HT I/O devices that are coupled to an HT I/O interconnect and return such information to the issuer of the transaction. Similarly, a type0 configuration write transaction stores information in the configuration registers that are located within HT I/O devices that are coupled to an HT I/O interconnect.

When the host processor 505 issues a type0 configuration read transaction on the HT I/O interconnect 540, the capabilities of the first HT switch 510 can be retrieved. Because the retrieved capabilities would indicate that an HT switch is coupled to I/O bus 540, the host processor 505 would issue one or more type0 read transactions that retrieve the number of ports in the first HT switch 510.

5.6.2 Define *Px Match* values

After the host processor 505 determines the number of ports on the first HT switch 510, the *Px Match* configuration-forwarding table elements that correspond to the enabled and disabled primary ports of the first HT switch 510 may be defined by the host processor 505 using type0 configuration write transactions. An enabled port is a port that is coupled to a device. A disabled port is a port that is not coupled to a device. For example, port P0 of the first HT I/O device 510 is enabled because it is coupled to the host processor 505 and port P1 is disabled because it is not coupled to a device.

In one embodiment, the value of *Temp-Primary-Segment, i.e.*, 0x1111, would be stored in *P0 Match* because port P0 is enabled. Thus, the first HT switch's port P0 is thereby associated with *Primary-Segment* 0x1111. *Temp-Primary-Segment* would then be decremented by one. The value of *Temp-Primary-Segment*, *i.e.*, 0x1110 would not be stored in *P1 Match* because port P1 is disabled. Instead, a value of 0x0000 would be stored in *P1 Match*. However, *Temp-Primary-Segment* would still be decremented by one. The process of storing values in *Px Match* for enabled primary ports and decrementing *Temp-Primary-Segment* for all primary ports, whether enabled or disabled, would continue for the other primary ports in the first HT switch 510.

5.6.3 Define *Sy Match* values

Next, the *Sy Match* configuration-forwarding table elements that correspond to the secondary ports of the first HT switch 510 may be similarly defined. The value of *Temp-Secondary-Segment, i.e.*, 0x1111, would be stored in *S0 Match* because port S0 is

enabled. Thus, the first HT switch's port S0 is thereby associated with *Secondary-Segment* 0x1111. *Temp-Secondary-Segment* would then be decremented by one. The value of *Temp-Secondary-Segment*, *i.e.*, 0x1110 would be stored in *S1 Match* because port S1 is enabled. This storage also associates port S1 with *Secondary-Segment* 0x1110. *Temp-Secondary-Segment* would then be decremented by one. The process of storing values in *Sy Match* for enabled secondary ports and decrementing *Temp-Secondary-Segment* for all secondary ports, whether enabled or disabled, would continue for the other secondary ports in the first HT switch 510.

5.6.4 Type1 Configuration Transactions

In one embodiment of the invention, in addition to the type0 configuration read and write transactions that may be issued by the host processor 505, the host processor 505 may also issue type1 configuration read and write transactions. Type1 configuration transactions may be used to configure devices on secondary segments. Type1 configuration transactions instruct the receiving device to forward the transaction to a secondary segment. When the type1 configuration transaction reaches its target bus, it is converted into a type0 configuration read or write transaction by an HT switch. In some embodiments of the invention, the type1 configuration transaction includes a first field, *HT-Primary-Segment*, which allows the host processor 505 to separately address different primary segments. In addition, the type1 configuration transaction may also include a second field, *HT-Secondary-Segment*, which allows the host processor 505 to separately address different secondary segments. By including both of these fields in type1 configuration transactions, the host processor 505 may address segments in an efficient

hierarchical numbering system. Such addressing is more efficient than addressing with a single "BusNumber" field that is utilized in the prior art.

5.7 Configuration of the Second HT Switch

After the *Px Match* and *Sy Match* elements in the first HT switch's configuration-forwarding table have been defined, the host processor 505 may utilize these elements to configure the second HT switch 515. In one embodiment of the invention, the host processor 505 may issue a type1 read transaction that reads the capabilities of the second HT switch 515 by issuing a type1 configuration read transaction. This transaction would include *HT-Primary-Segment* equal to 0x1111 and *HT-Secondary-Segment* equal to 0x1111. The first HT switch 510 would receive the type1 configuration read transaction. Because *P0 Match* is equal to *HT-Primary-Segment*, and *S0 Match* is equal to *HT-Secondary-Segment*, the first HT switch 510 would convert the type1 configuration transaction into a type0 configuration transaction, and would forward the type0 configuration transaction, via port S0, to the second HT switch 515. (See Section 5.9.2 for other methods of forwarding type1 configuration transactions.) When the second HT switch 515 receives the type0 configuration read transaction, it will provide, via the first HT switch 510, the capabilities of the second HT switch 515. The host processor 505 would then issue type1 configuration write transactions, which would be converted into type0 configuration write transactions to define the elements in the configuration-forwarding table in the second HT switch 515 as discussed above. After configuration, in one embodiment of the invention, the second HT switch's configuration-forwarding table

would contain the following values: *P0-Match* = 0x1101; *P1-Match* = 0x0000; *S0-Match* = 0x1101; and *S1-Match* = 0x1100.

5.8 Configuration of the Third HT Switch

The host processor may also issue a type1 read transaction that reads the capabilities of the third HT switch 525 by issuing a type1 configuration read transaction that includes *HT-Primary-Segment* equal to 0x1111 and *HT-Secondary-Segment* equal to 0x1110. The first HT switch 510 would receive the type1 configuration read transaction. Because *P0 Match* is equal to *HT-Primary-Segment*, and *S1 Match* is equal to *HT-Secondary-Segment*, the first HT switch 510 would convert the type1 configuration transaction into a type0 configuration transaction, and would forward the type0 configuration transaction, via port S1, to the third HT switch 525. The third HT switch 525 could then be configured so that its configuration-forwarding table would contain the following values: *P0-Match* = 0x1011; *P1-Match* = 0x0000; *S0-Match* = 0x1011; and *S1-Match* = 0x1010.

5.9 Preparing to Configure the First HT I/O Device

In order to configure the first HT I/O device 520, the host processor needs to issue configuration transactions that traverse both the first HT switch 510 and the second HT switch 515 and then reach the first HT I/O device 520. Thus, in one embodiment of the invention, the host processor may issue a type1 configuration transaction, which is forwarded as a type1 configuration transaction from the first HT switch 510 to the second HT switch 515. The second HT switch 515 would then convert the type1 configuration

transaction into a type0 configuration transaction and forward the type0 configuration transaction to the first HT I/O device 520.

5.9.1 Defining the Mask values in the First HT Switch

In one embodiment of the invention, the host processor may store *Mask* elements in the configuration-forwarding table so that the elements may be later utilized to determine whether to forward a received type1 configuration transaction as a type0 configuration transaction or as a type1 configuration transaction. In such embodiments, after the third HT switch 525 has been configured as discussed in Section 5.6, the host processor may issue type0 configuration write transactions to define the *Px Mask* elements and *Sy Mask* elements in the first HT switch's configuration-forwarding tables. In one embodiment, a value of 0x1111 may be placed in all *Px Mask* elements that correspond to primary ports that are enabled and a value of 0x0000 may be stored in all *Px Mask* elements that correspond to primary ports that are disabled. Thus, *P0-Mask* would be set to 0x1111 because primary port P0 is enabled and *P1-Mask* would be set to 0x0000 because primary port P1 is disabled. Similarly, a value of 0x1111 may be placed in all *Sy Mask* elements that correspond to secondary ports that are enabled and a value of 0x0000 may be stored in all *Sy Mask* elements that correspond to secondary ports that are disabled. Thus, *S0-Mask* and *S1-Mask* would both be set to 0x1111 because both secondary ports S0 and S1 are enabled.

5.9.2 Forwarding Type1 Configuration Transactions as Type1 Configuration Transactions

After the above *Mask* elements are stored in the first HT switch's configuration-forwarding table, the first HT switch 510 can then utilize the *Mask* elements to determine whether to forward received type1 configuration transactions as either type0 configuration transactions or type1 configuration transactions. In one embodiment, the first HT switch 510 would forward a type1 transaction that was received on primary port "*x*" to secondary port "*y*" as a type0 transaction when the following logical equation is true:

$$((HT\text{-}Primary\text{-}Segment \;\&\; Px\ Mask) == (Px\ Match \;\&\; Px\ Mask)) == 1 \;\&\&$$

$$(HT\text{-}Secondary\text{-}Segment == Sy\ Match) == 1$$

In this embodiment, the first HT switch 510 would forward a type1 transaction that was received on primary port "*x*" to secondary port "*y*" as a type1 transaction when the following logical equation is true:

$$((HT\text{-}Primary\text{-}Segment \;\&\; Px\ Mask) == (Px\ Match \;\&\; Px\ Mask)) == 1 \;\&\&$$

$$((HT\text{-}Secondary\text{-}Segment \;\&\; Sy\ Mask) == (Sy\ Match \;\&\; Sy\ Mask)) == 1$$

5.10 Configuring the First HT I/O Device

As discussed in Section 5.9.2, after storing *Mask* elements in the first HT switch's configuration-forwarding table, the host processor 505 can issue type1 configuration transactions that allow reading and writing to configuration registers in the first HT I/O device 520. Thus, the host processor 505 can assign the first HT I/O device a Unit ID.

5.11 Configuration of the Fourth HT Switch

Using the methods discussed above, the fourth HT switch's configuration-forwarding table *Match* values could be defined as follows: *P0 Match* = 0x1011; *P1 Match* = 0x0000; *S0 Match* = 0x0110; and *S1 Match* = 0x0000.

5.12 Configuration of the Second HT I/O Device

Using the methods discussed above, the third HT I/O switch's configuration-forwarding table *Mask* values could be defined as follows: *P0 Mask* = 0x1111; *P1 Mask* = 0x0000; *S0 Mask* = 0x1111; and *S1 Mask* = 0x0000. After such *Mask* values are defined, the host processor 505 can then issue type1 configuration transactions that may be forwarded to the second HT I/O device 535 as type0 configuration transactions.

Thus, the second HT I/O device 535 may be assigned a Unit ID as discussed in Section 5.10.

5.13 Packet-Forwarding Table

In addition to the configuration-forwarding tables discussed above, the HT switches may also include a packet-forwarding table. The packet-forwarding table may be used to forward packets from primary HT segments to secondary HT segments through HT switches. One embodiment of a packet-forwarding table 600 is presented in Figure 6. The first element in the packet-forwarding table may be identical to the capability ID 205 discussed in Section 5.4.1. Similarly, the second element in the packet-forwarding table, the capabilities pointer, may be identical to the capabilities pointer 210 described in Section 5.4.2.

5.13.1 Command Bits

The third element in the packet-forwarding table 600 may contain command bits. A first group of the command bits may indicate the number of enabled primary ports present in the HT switch. A second group of the command bits may indicate the number of enabled secondary ports present in the HT switch. The command bits may also include a third group of bits that indicate the layout of the remainder of the packet-forwarding table.

5.13.2 Packet-Forwarding Data

The remainder of the packet-forwarding table contains packet-forwarding data. The packet-forwarding data includes segment numbers for primary segments and secondary segments. These segments have reachable HT devices connected to them. As

shown in Figure 6, the packet-forwarding table may be indexed by Unit IDs. As discussed below, the packet-forwarding data may be utilized by the computer system to forward packets from primary ports to secondary ports and to forward packets from secondary ports to primary ports.

Figure 7 presents a portion of a computer system 700. As can be seen in Figure 7, an HT switch 705 is coupled to two HT I/O devices. The HT switch 705 is also coupled to other devices in the computer system via its primary ports P0 and P1. These devices are not shown. The first HT I/O device 730 has previously been assigned Unit ID 0. This device is coupled to the HT switch's S0 port. The second HT I/O device 715 has previously been assigned Unit ID 1. The second I/O device 715 is coupled to the HT switch's S1 port.

Figure 8 presents a packet-forwarding table 800. Packet-forwarding table 800 contains data that may be utilized by a computer system to forward packets from the HT switch's primary ports to the HT switch's secondary ports. For each Unit ID, the table contains a *Primary-Segment* value and a *Secondary-Segment* value. These values may be determined from the data obtained during the generation of the configuration-forwarding table.

One method of utilizing the packet-forwarding table 800 to forward packets follows. When the HT switch 705 receives a packet, the HT switch determines the Unit ID of the incoming packet. Next, the HT switch 705 looks up the Unit ID in the packet-forwarding table. Then the HT switch 705 retrieves the *Primary-Segment* value and the *Secondary-Segment* value. Finally, the retrieved values are utilized to identify the outgoing port and forward the packet.

For example, element 805, *Primary-Segment* = 1, and element 810, *Secondary-Segment* = 0, define a path through HT switch 705 for packets with a Unit ID equal to 0. Thus, any packet received with a unit ID equal to "0" from *Primary- Segment 1* 725 would always be forwarded to *Secondary-Segment 0* 730. Similarly element 815, *Primary-Segment* = 0, and element 820, *Secondary-Segment* = 1, define a path through HT switch 705 for packets with a Unit ID equal to 1. Thus, any packet received with a unit ID equal to "1" from *Primary-Segment 0* 720 would always be forwarded to *Secondary-Segment 1* 715.

Element 805, 810, 815, and 820 may also be utilized to define a path from the HT switch's secondary ports to the HT switch's primary ports. Thus, any packet received with a unit ID equal to "0" would always be forwarded to *Primary-Segment 1* 725. Similarly, any packet received with a unit ID equal to "1" would always be forwarded to *Primary-Segment 0* 720. Thus, the packet-forwarding table may be utilized to efficiently forward packets through HT switches.

5.14 Conclusion

The foregoing descriptions of embodiments of the present invention have been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. For example, the above description discusses computer systems with one or more HT I/O interconnectes. However, the invention is not limited to such computer systems. The invention may be utilized on computer systems with other types of buses. Accordingly, many modifications and variations may be apparent to practitioners skilled in the art. For

example, Figures 9 through 16 are alternative embodiments of the invention.

Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended claims.